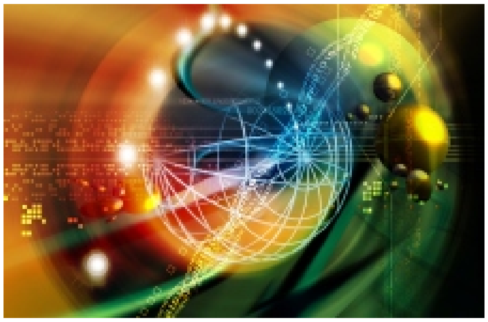
# Adaptive software -- a late bloomer

August 20 2010



Adaptive software is the largely unfulfilled promise of mobile technology, but now a new platform developed by European researchers promises to finally deliver software that reconfigures itself depending on the context.

Adaptive software - software that can configure itself automatically to deliver the maximum functionality for a given context - is finally coming. One of the long-promised fruits of the mobile technology tree, adaptation has proven a late bloomer, in large part due to the number of extremely tough problems that dog the idea.

For starters, there are thousands of different potential devices, from mobile phones to media players to computers. Added to that are the even vaster number of applications.

“The MUSIC project wanted to create a platform for adaptive applications that could work on any device and with any software,” explains Geir Horn, researcher with SINTEF in Norway and coordinator of the project. “[It’s] because people often want software to work differently in different contexts.”

## Faulty defaults

For example, most emailing software downloads each email with all attachments in the order they were received. On a slow internet connection, it can mean waiting an hour or more to retrieve all the emails a user has received, and the connection could even drop out before the important last mail has been downloaded.

But email clients have a ‘slow connection’ setting that lets users download just the basic information: the subject, the sender and the size of the file, and it can be very useful when roaming data feeds or when the bandwidth is limited.

And it is not just download options that can be optimised for particular contexts. Encryption, too, can be usefully adjusted for office use or when on the move, with simple or even no encryption for the office, but extremely strong encryption outside the office.

Functions like these are typically buried behind obscure menu items and several dialogue boxes, and there are literally hundreds of possible options for all the different software on one device, and even more options for other devices. Users would quickly become overwhelmed if they have to change settings manually.

“Of course, the email software developers could write their software so that it adapts when the internet connection is slow, or when the user is moving, or when the user is in a particular location, but there are possibly hundreds of different situations that they would need to think of, and it quickly gets prohibitively expensive," stresses Horn. "And that is just for one application."

MUSIC handles all these issues. It takes account of a user's location, whether it is at home, the office, in a car or on public transport. It notes the device in use, whether a mobile phone, netbook or desktop, and it is aware of dozens of other variables, from the strength of the internet connection to the time of day, from light levels to the condition of the battery.

## Good guesses

By combining all this information, the MUSIC middleware can make a very good guess of the users' context, what the user will want to do at a particular time, in a specific location with a given device.

The MUSIC platform does not need to be configured by the user, it derives its possible settings from the high-level behaviour descriptions that are part of most software developed today. High-level behaviours describe functions such as displaying SMS messages via text or using audio.

These descriptions tell the MUSIC software what settings can be altered, and MUSIC then simply uses the most appropriate settings for a given context. In the SMS example, it would render the message in text when the user is in a restaurant, but render it in audio when he or she is in a car.

"The biggest challenge that the MUSIC consortium faced was trying to create software that would install on any device, whether it is a mobile phone, PDA, netbook or server," Horn relates. "In the end, we decided to use Java, which we thought was the most appropriate cross-platform development language."

The developers also used a plugin paradigm so the software can be extended in useful ways. For example, any software can take advantage of MUSIC functionality simply by developing a plugin for the platform. Similarly, the platform can expand its functionality via plugins too.

So while the base MUSIC configuration may take account of location via GPS and time via the device clock, plugins could expand the base configuration to take advantage of other sensors, such as a compass, accelerometer or light meter.

## Mission accomplished

The MUSIC project achieved all its design goals and has created a robust prototype platform. The software is designated open source and is freely available. There is a fledgling development community coalescing around the platform and, while there is no formal commercialisation plan as such, it is very likely that MUSIC will slowly, almost surreptitiously, become part of the mobile landscape.

That is because the platform is tiny, just a few hundred kilobytes in its base configuration. "So if a company wants to offer services that depend on MUSIC, they can just bundle the software with the service," Horn remarks.

He offers the example of a tourist office in, say, Turin. The tourism office can develop a package that will show users specific information at a precise location - say the history of a famous church - when the user is standing there.When the user subscribes to the service, the software can be downloaded to their phone directly. That makes the MUSIC functionality available to any other MUSIC-enabled software on the user's phone.

By degrees and utility, then, MUSIC has a very strong chance of becoming a very common platform and then, finally, mobile technology will evolve to become truly ‘adaptive’.

Check out the plugins repository on the MUSIC website.

**More information:** MUSIC project - developer.berlios.de/projects/ist-music/

Provided by ICT Results

Citation: Adaptive software -- a late bloomer (2010, August 20) retrieved 26 April 2024 from https://phys.org/news/2010-08-software-late-bloomer.html